APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/705,208	11/10/2003	Aaron Joseph McBride	2966-031366	2941

28289   7590   03/04/2010
THE WEBB LAW FIRM, P.C.
700 KOPPERS BUILDING
436 SEVENTH AVENUE
PITTSBURGH, PA 15219

EXAMINER
BUSS, BENJAMIN J

ART UNIT	PAPER NUMBER
2129	

MAIL DATE	DELIVERY MODE
03/04/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Application Number: 10/705,208
Filing Date: November 10, 2003
Appellant(s): MCBRIDE ET AL.

Alexander Detschelt (Reg. No. 50,261)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 12/18/2009 appealing from the Office action mailed

11/14/2008.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

2003/0163783   CHIKIRIVAO et al.   8-2003

WALLACE, R.S. "The Elements of AIML Style" ALICE A.I. Foundation, Inc. (March 28, 2003), pp. 1-86

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1, 13-15, 23, 24, 27-35 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chikirivao** (USPAP 2003/0163783) in view of **Wallace** ("The Elements of AIML Style").

Independent Claim 1:

Chikirivao teaches:

- providing a template interface to the administrator, wherein the template includes at least one field to elicit information from the administrator (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository" ¶38-39),

- receiving information from the administrator into the template (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository" ¶38-39 and "entering and saving of data into a template" ¶43), and

- making the information accessible to a rules-based program for use in providing the at least one response in reply to a request from a user (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule" ¶38-39 and "rules created by an administrator are preferably saved in the rule repository" ¶29), wherein the step of making the information accessible to the rules-based program saves the information as part of the template into rules (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository" ¶38-39 and "entering and saving of data into a template" ¶43 and "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule" ¶38-39 and "rules created by an administrator are preferably saved in the rule repository" ¶29 and "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule" ¶38-39 and "rules created by an administrator are preferably

saved in the rule repository" ¶29; *The rule repository is clearly structured data storage*), and

wherein the step of saving the information into rules includes the steps of:

- o retrieving rules (pages 1-7 especially "system obtains the rule" ¶38-39),

- o for each rule retrieved, determining whether the rule needs information (pages 1-7
     especially "administrator may need to specify more or less information" ¶40 and "rules
     which are generated based upon … information … based upon parameters specified"
     ¶31 and "information and/or sub-rules needed to make such determinations" ¶32 and
     "rules may be designed with any level of interactivity and/or user knowledge required and
     may include and utilize data and other information" ¶33 and "extracts from the provided
     information those parameters required by the rule(s)" ¶44-48), and

- o if the rule needs information, retrieving the information from a corresponding field in the
     template and inserting the information into the rule (pages 1-7 especially "extracts from
     the provided information those parameters required by the rule(s)" ¶44-48 and
     "administrator may need to specify more or less information" ¶40 and "routing of
     information based upon the input template" ¶59),

- o wherein the step of determining whether the rule needs information includes determining
     if either a response layer or a logic layer needs information (pages 1-7 especially "rule
     which requires the user to provide inputs as to specific needs" ¶35 and "administrator
     may need to specify more or less information" ¶40 and "rules which are generated based
     upon … information … based upon parameters specified" ¶31 and "information and/or
     sub-rules needed to make such determinations" ¶32 and "rules may be designed with
     any level of interactivity and/or user knowledge required and may include and utilize data
     and other information" ¶33 and "extracts from the provided information those parameters
     required by the rule(s)" ¶44-48), and

     - ▪ retrieving information indicated as needed from a corresponding field in the
          template and inserting the information into the response layer or logic layer, as
          called for by the signifier (pages 1-7 especially "templates and other features that

enable a user to expeditiously enter the necessary information required" ¶43 and "extracts from the provided information those parameters required by the rule(s)" ¶44-48 and "information is received from the field" ¶50 and "routing of information based upon the input template" ¶59 and "administrator may need to specify more or less information" ¶40);

Chikirivao fails to teach:

- wherein the step of determining whether a layer needs information, includes the step of identifying a signifier in the layer,
  - o wherein the signifier is an identifier configured to call for information such that the call for information invokes a process to select in the information from a corresponding field in the template so that the information will be linked to the rule, and
  - o wherein the logic layer is configured to choose between various responses provided by the administrator,
  - o wherein at least one of the responses is recognized by the logic layer,
  - o wherein the chosen response is the response to be used in the response layer.

Wallace teaches:

  - o if the rule needs information, retrieving the information from a corresponding field in the template and inserting the information into the rule (p1-83 especially i.e. p12-13 or p21-23 or p38-39 or p53-57),
    - ▪ wherein the step of determining whether the rule needs information includes determining if either a response layer or a logic layer needs information by identifying the presence of a signifier in the response layer or the logic layer, respectively, wherein the signifier is an identifier configured to call for information such that the call for information invokes a process to select the information from a corresponding field in the template so that the information will be linked to the rule (p1-83 especially "AIML tags transform the reply into a mini computer

> program which can save data, activate other programs, give conditional
>
> responses, and recursively call the pattern matcher to insert the responses from
>
> other categories" p12-13 or p53-55; *Examiner points out that AIML tags indicate*
>
> *when the rules need more information. These tags can be part of the logic layer*
>
> *used to determine the appropriate branches to take in seeking a response for the*
>
> *input, or they can be part of the response layer used to give responses or call*
>
> *processes in response to the input*), and
>
> ▪ wherein the logic layer is configured to choose between various responses
>
>   provided by the administrator (p1-83 especially i.e. p53-55 or p56-57),
>
> ▪ wherein at least one of the responses is recognized by the logic layer (p1-83
>
>   especially i.e. p21-23 or p52-53; *Examiner points out that AIML relies on the*
>
>   *graph master / logic layer recognizing which administrator provided response is*
>
>   *appropriate for the given input*),
>
> ▪ wherein the chosen response is the response to be used in the response layer
>
>   (p1-83 especially i.e. "The algorithm finds best-matching pattern for each input.
>
>   The category ties the response template directly to the stimulus pattern" p38-39
>
>   or p56-57), and
>
> ▪ retrieving information indicated as needed from a corresponding field in the
>
>   template and inserting the information into the response layer or logic layer, as
>
>   called for by the signifier (p1-83 especially i.e. p12-13 or p16-17 or p56-57);

Rationale:

> **Chikirivao** and **Wallace** are from the same field of endeavor, information management and
>
> retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention
>
> to modify the teachings of **Chikirivao** by identifying a signifier / tag in a layer such as the
>
> response or logic layer to determine whether information is needed as taught by **Wallace** for the
>
> benefit of making it easy for non-programmers to simply / easily write interactive responses for a
>
> variety of applications (**Wallace** p36, 77-80).

Independent Claim 35:

Chikirivao teaches:

- An interface configured to receive information from the administrator (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository" ¶38-39 and "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository" ¶38-39 and "entering and saving of data into a template" ¶43);

- A template accessible to the administrator, wherein the template includes at least one field to elicit information from the administrator (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository" ¶38-39 and "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository" ¶38-39 and "entering and saving of data into a template" ¶43);

- An engine configured to:

   o Make the information accessible to a rules-based program that provides the at least one response in reply to the inputs from the user (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule" ¶38-39 and "rules created by an administrator are preferably saved in the rule repository" ¶29);

   o Save the information as part of the template into rules (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule" ¶38-39 and "rules created by an administrator are preferably saved in the rule repository" ¶29).

   o Retrieve the rules (pages 1-7 especially "system obtains the rule" ¶38-39);

   o For each rule retrieved, determine whether the rule needs information (pages 1-7 especially "administrator may need to specify more or less information" ¶40 and "rules

which are generated based upon … information … based upon parameters specified"

¶31 and "information and/or sub-rules needed to make such determinations" ¶32 and

"rules may be designed with any level of interactivity and/or user knowledge required and

may include and utilize data and other information" ¶33 and "extracts from the provided

information those parameters required by the rule(s)" ¶44–48);

o Retrieve the information from a corresponding field in the template and insert the

information into the rule if the rule needs information (pages 1-7 especially "extracts from

the provided information those parameters required by the rule(s)" ¶44–48 and

"administrator may need to specify more or less information" ¶40 and "routing of

information based upon the input template" ¶59);

o Determine if either a response layer or a logic layer needs information (pages 1-7

especially "rule which requires the user to provide inputs as to specific needs" ¶35 and

"administrator may need to specify more or less information" ¶40 and "rules which are

generated based upon … information … based upon parameters specified" ¶31 and

"information and/or sub-rules needed to make such determinations" ¶32 and "rules may

be designed with any level of interactivity and/or user knowledge required and may

include and utilize data and other information" ¶33 and "extracts from the provided

information those parameters required by the rule(s)" ¶44–48); and

▪ retrieve information indicated as needed from a corresponding field in the

template and inserting the information into the response layer or logic layer, as

called for by the signifier (pages 1-7 especially "templates and other features that

enable a user to expeditiously enter the necessary information required" ¶43 and

"extracts from the provided information those parameters required by the rule(s)"

¶44–48 and "information is received from the field" ¶50 and "routing of information

based upon the input template" ¶59 and "administrator may need to specify more

or less information" ¶40);

Chikirivao fails to teach:

- Wherein the step of determining whether a layer needs information, includes identifying a signifier in the layer,

  o wherein the signifier is an identifier configured to call for information such that the call for information invokes a process to select in the information from a corresponding field in the template so that the information will be linked to the rule, and

  o wherein the logic layer is configured to choose between various responses provided by the user,

  o wherein at least one of the responses is recognized by the logic layer,

  o wherein the chosen response is the response to be used in the response layer.

Wallace teaches:

  o determine if either a response layer or a logic layer needs information by identifying the presence of a signifier in the response layer or the logic layer, respectively, wherein the signifier is an identifier configured to call for information such that the call for information invokes a process to select the information from a corresponding field in the template so that the information will be linked to the rule (p1-83 especially i.e. "AIML tags transform the reply into a mini computer program which can save data, activate other programs, give conditional responses, and recursively call the pattern matcher to insert the responses from other categories" p12-13 or p53-55; *Examiner points out that AIML tags indicate when the rules need more information. These tags can be part of the logic layer used to determine the appropriate branches to take in seeking a response for the input, or they can be part of the response layer used to give responses or call processes in response to the input*), and

  o wherein the logic layer is configured to choose between various responses provided by the administrator (p1-83 especially i.e. p53-55 or p56-57),

     o   wherein at least one of the responses is recognized by the logic layer (p1-83 especially

         i.e. p21-23 or p52-53; *Examiner points out that AIML relies on the graph master / logic*

         layer recognizing which administrator provided response is appropriate for the given

         input),

     o   wherein the chosen response is the response to be used in the response layer (p1-83

         especially i.e. "The algorithm finds best-matching pattern for each input. The category

         ties the response template directly to the stimulus pattern" p38-39 or p56-57), and

     o   retrieve information indicated as needed from a corresponding field in the template and

         inserting the information into the response layer or logic layer, respectively (p1-83

         especially i.e. p12-13 or p16-17 or p56-57);

<u>Rationale</u>:

     Chikirivao and **Wallace** are from the same field of endeavor, information management and

     retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention

     to modify the teachings of **Chikirivao** by identifying a signifier / tag in a layer such as the

     response or logic layer to determine whether information is needed as taught by **Wallace** for the

     benefit of making it easy for non-programmers to simply / easily write interactive responses for a

     variety of applications (**Wallace** p36, 77-80).

Dependent claims 13-15, 22-34, and 40:

Claim 13:

Chikirivao fails to teach:

Wherein the signifier is a tag in a text string. Wallace teaches: Wherein the signifier is a tag in a text string

(pl-83 especially i.e. "AIML tags transform the reply into a minicomputer program which can save data,

activate other programs, give conditional responses, and recursively call the pattern matcher to insert the

responses from other categories" p12-13 or p15 or p40 or p53-55).

<u>Rationale:</u>

Chikirivao and Wallace are from the same field of endeavor, information management and retrieval. It

would have been obvious to one of ordinary skill in the art at the time of the invention to modify the

teachings of Chikirivao by identifying a signifier / tag in a text string to determine whether information is

needed as taught by Wallace for the benefit of making it easy for non-programmers to simply / easily write

interactive responses for a variety of applications (Wallace p36, 77-80).

Claim 14:

Chikirivao fails to teach:

Wherein the signifier is an instruction embedded in a text string.

Wallace teaches: Wherein the signifier is an instruction embedded in a text string (p I-83 especially i.e.

p12-13 or p15-18).

Rationale:

Chikirivao and Wallace are from the same field of endeavor, information management and retrieval. It

would have been obvious to one of ordinary skill in the art at the time of the invention to modify the

teachings of Chikirivao by identifying a instruction embedded in a text string to determine whether

information is needed as taught by Wallace for the benefit of making it easy for non-programmers to

simply / easily write interactive responses for a variety of applications (Wallace p36, 77-80).

Claim 15:

Chikirivao fails to teach:

Wherein the signifier is a code.

Wallace teaches: Wherein the signifier is a code (p I-83 especially i.e. "<system> tag executes any

program accessible as an operating system shell command, and inserts the results in the reply. Similarly,

the <javascript> tag allows arbitrary scripting" p12-13 or p15-18).

Rationale:

Chikirivao and Wallace are from the same field of endeavor, information management and retrieval. It

would have been obvious to one of ordinary skill in the art at the time of the invention to modify the

teachings of Chikirivao by identifying a code to determine whether information is needed as taught by

Wallace for the benefit of making it easy for non-programmers to simply / easily write interactive

responses for a variety of applications (Wallace p36, 77-80).

Claim 23:

Chikirivao teaches:

Wherein the step of making the information accessible to the rules-based program is accomplished by

receiving a manual command from a user (pages 1-7 especially "access to a rule may be specified

manually or automatically" (para. 43).

Claim 24:

Chikirivao teaches:

Wherein the step of making the information accessible to the rules-based program is accomplished

automatically upon the occurrence of a predefined event (pages 1-7 especially "access to a rule may be

specified manually or

automatically" (para. 43).

Claim 27:

Chikirivao teaches:

Wherein the predefined event is activation of a save function by the administrator (pages 1-7 especially

"access to a rule may be specified manually or automatically ... rule may be activated upon the entering

and saving of data into a template" (para. 43).

Claims 28 and 40:

Chikirivao teaches:

Further including the step of enabling the administrator to edit the information (pages 1-7 especially

"querying the administrator ... modify an existing rule ... create a customized rule based upon a pre-

existing customizable rule template saved in the rule repository ... allow the user to modify/customize the

rule" (para. 38-39) and "enables such administrators to ... edit ... the rules" (para. 29).

Claim 29:

Chikirivao teaches:

Wherein the step of enabling the administrator to edit the information includes the steps of:

Retrieving the information (pages 1-7 especially "obtains the rule and provides those interfaces necessary to allow the user to modify/customize the rule" (para. 38-39),

Posting the information in at least one appropriate field in the template (pages 1-7 especially "based upon a pre-existing customizable rule template ... obtains the rule and provides those interfaces necessary to allow the user to modify/customize the rule" (para. 38-39) and "templates and other features that enable a user to expeditiously enter the necessary information required for a given task" (para. 43),

Receiving edited information from the administrator into the template (pages 1-7 especially "based upon a pre-existing customizable rule template ... obtains the rule and provides those interfaces necessary to allow the user to modify/customize the rule" (para. 38-39) and "templates and other features that enable a user to expeditiously enter the necessary information required for a given task" (para. 43), and

Making the edited information accessible to the rules-based program for use in providing the at least one response in reply to a request from the user (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule" (para. 38-39) and "rules created by an administrator are preferably saved in the rule repository" (para. 29).

Claim 30:

Chikirivao teaches wherein:

The step of making the information accessible to the rules-based program saves the information as part of the template (pages 1-7 especially "access to a rule may be specified manually or automatically ... rule may be activated upon the entering and saving of data into a template" (para. 43), and The step of retrieving the information includes the step of restoring the information to the at least one field (pages 1-7

especially "based upon a pre-existing customizable rule template ... obtains the rule and provides those interfaces necessary to allow the user to modify/customize the rule" (para. 38-39).

Claim 31:

Chikirivao teaches wherein:

The step of making the information accessible to the rules-based program saves the information as structured data (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule" (para. 38-39) and "rules created by an administrator are preferably saved in the rule repository" (para. 29); The rule repository is clearly structured data storage).

Chikirivao fails to teach wherein:

The step of retrieving the information includes the steps of, for at least one of the at least one field in the template:

Retrieving instructions indicating where the information is stored, and

Executing the instructions to retrieve the information.

Wallace teaches wherein:

The step of making the information accessible to the rules-based program saves the information as structured data (pl-83 especially i.e. "structure" p21-24 or p30 or p50 or p79), and

The step of retrieving the information includes the steps of, for at least one of the at least one field in the template:

Retrieving instructions indicating where the information is stored (pl-83 especially i.e. p21-24)

Executing the instructions to retrieve the information (p l-83 especially i.e. p12-13 or p21-24 or p38-39 or p56-57).

<u>Rationale:</u>

Chikirivao and Wallace are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Chikirivao by using executing instructions to retrieve information is needed as taught by

Wallace for the benefit of making it easy for non-programmers to simply / easily write interactive

responses for a variety of applications (Wallace p36, 77-80).

Claim 32:

Chikirivao teaches wherein:

The step of making the information accessible to the rules-based program saves the information into rules

(pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing

customizable rule template saved in the rule repository ... either testing the rule or saving the rule" (para.

38-39) and "rules created by an administrator are preferably saved in the rule repository" (para. 29).

Chikirivao fails to teach wherein:

The step of retrieving the information includes the steps of, for at least one of the at least one field in the

template:

Retrieving instructions indicating where the information is stored, and

Executing the instructions to retrieve the information.

Wallace teaches wherein:

The step of making the information accessible to the rules-based program saves the information as

structured data (pl-83 especially i.e. "structure" p21-24 or p30 or p50 or p79), and

The step of retrieving the information includes the steps of, for at least one of the at least one field in the

template:

Retrieving instructions indicating where the information is stored (p l-83 especially i.e. p21-24)

Executing the instructions to retrieve the information (p l-83 especially i.e. p12-13 or p21-24 or p38-39 or

p56-57).

Rationale:

Chikirivao and Wallace are from the same field of endeavor, information management and retrieval. It

would have been obvious to one of ordinary skill in the art at the time of the invention to modify the

teachings of Chikirivao by using executing instructions to retrieve information is needed as taught by

Wallace for the benefit of making it easy for non-programmers to simply / easily write interactive

responses for a variety of applications (Wallace p36, 77-80).

Claim 33:

Chikirivao teaches wherein:

The step of making the information accessible to the rules-based program saves the information into rules

(pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing

customizable rule template saved in the rule repository ... either testing the rule or saving the rule" (para.

38-39) and "rules created by an administrator are preferably saved in the rule repository" (para. 29).

Chikirivao fails to teach wherein:

The step of retrieving the information includes the steps of, for each rule used:

Determining whether the rule includes a signifier, and

If a signifier is included, executing instructions from the signifier to retrieve the information associated with

the rule.

Wallace teaches wherein:

The step of retrieving the information includes the steps of, for each rule used:

Determining whether the rule includes a signifier (p 1-83 especially i.e. p12-13 or p21-23 or p38-39 or

p53-57), and

If a signifier is included, executing instructions from the signifier to retrieve the information associated with

the rule (p 1-83 especially i.e. p12-13 or p21-23 or p38-39 or p53-57).

Rationale:

Chikirivao and Wallace are from the same field of endeavor, information management and retrieval. It

would have been obvious to one of ordinary skill in the art at the time of the invention to modify the

teachings of Chikirivao by using executing instructions to retrieve information is needed as taught by

Wallace for the benefit of making it easy for non-programmers to simply / easily write interactive

responses for a variety of applications (Wallace p36, 77-80).

Claim 34:

Chikirivao teaches wherein:

The step of making the information accessible to the rules-based program saves the information into rules (pages 1-7 especially "querying the administrator ... create a customized rule based upon a pre-existing customizable rule template saved in the rule repository ... either testing the rule or saving the rule" (para. 38-39) and "rules created by an administrator are preferably saved in the rule repository" (para. 29).

Chikirivao fails to teach wherein:

The step of retrieving the information includes the steps of, for each rule used:

Determining whether the rule includes a signifier, and

If a signifier is included, retrieving the information tagged in the rule.

Wallace teaches wherein:

The step of retrieving the information includes the steps of, for each rule used:

Determining whether the rule includes a signifier (p 1-83 especially i.e. p12-13 or p21-23 or p38-39 or p53-57), and

If a signifier is included, executing instructions from the signifier to retrieve the information associated with the rule (p 1-83 especially i.e. p12-13 or p21-23 or p38-39 or p53-57).

<u>Rationale:</u>

Chikirivao and Wallace are from the same field of endeavor, information management and retrieval. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the teachings of Chikirivao by using executing instructions to retrieve information is needed as taught by Wallace for the benefit of making it easy for non-programmers to simply / easily write interactive responses for a variety of applications (Wallace p36, 77-80).

<u>**Claim Rejections - 35 USC § 103**</u>

Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chikirivao (USPAP

2003/0163783) and Wallace ("The Elements of AIML Style") in view of Jammes (USPN 6,484,149).

Claim 22:

The combination of Chikirivao and Wallace fails to teach:

Wherein the step of retrieving rules retrieves all of the rules in a template information script.

Jammes teaches:

Wherein the step of retrieving rules retrieves all of the rules in a template information script (C1-56

especially "based on a template ... scripts to extract stored ... patterns ... against customization rules"

C43:40-65).\

Motivation:

Jammes and the combination of Chikirivao and Wallace are from the same field of endeavor, software

5 development. It would have been obvious to one of ordinary skill in the art at the time of the invention to

modify the combined teachings of Chikirivao and Wallace by retrieving all of the rules in a template

information script as taught by Jammes for the benefit of making the on-line experience more convenient

and expedient as well as more pleasant (Jammes C4:10-35).

Claim Rejections - 35 USC § 103

Claims 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chikirivao (USPAP

2003/0163783) and Wallace ("The Elements of AIML Style") in view of Habraken ("Microsoft Office XP 8-

in-1" - Part II1: Word - Chapter 2: Working with Documents).

Claim 25:

The combination of Chikirivao and Wallace fails to teach:

Wherein the predefined event is closing of the template.

Habraken teaches:

Wherein the predefined event is closing of the template (pages 4-16 especially "Before closing ... asks

whether you want to save these changes before closing" page 15).

<u>Motivation:</u>

Habraken and the combination of Chikirivao and Wallace are from the same field of endeavor, software. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the combined teachings of Chikirivao and Wallace by saving information to be available occurs when closing the template being edited as taught by Habraken for the benefit of not wanting to lose any recent changes (Habraken page 15) since you don't want to lose your valuable documents as you create them (Habraken page 13).

Claim 26:

The combination of Chikirivao and Wallace fails to teach:

Wherein the predefined event is passage of a predetermined amount of time.

Habraken teaches:

Wherein the predefined event is passage of a predetermined amount of time (pages 4-16 especially "AutoSave feature ... AutoRecoverInfo Every ... set the time interval between autosaves" page 13).

<u>Motivation:</u>

Habraken and the combination of Chikirivao and Wallace are from the same field of endeavor, software. It would have been obvious to one of ordinary skill in the art at the time of the invention to modify the combined teachings of Chikirivao and Wallace by saving information occurs after a predetermined amount of time as taught by Habraken for the benefit of not wanting to lose any recent changes (Habraken page 15) since you don't want to lose your valuable documents as you create them, so if you are really absent-minded about periodically saving your work, use the AutoSave feature (Habraken page 13).

Please note that as Appellant stated in §III of the instant Appeal Brief (page 2), "Only independent claims

1 and 35 are the subject of this Appeal." However, in the interest of being comprehensive, the detailed

rejections of dependent claims 13-15, 22-34, and 40 have been included in this Examiner's Answer since

these claims remain rejected on the same grounds as detailed in the Final Office Action mailed

11/14/2008:

- Claims 13-15, 23-24, 27-34, and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable

   over **Chikirivao** (USPAP 2003/0163783) and **Wallace** ("The Elements of AIML Style");

- Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Chikirivao** (USPAP

   2003/0163783) and **Wallace** ("The Elements of AIML Style") in view of **Jammes** (USPN

   6,484,149); and

- Claims 25-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Chikirivao**

   (USPAP 2003/0163783) and **Wallace** ("The Elements of AIML Style") in view of **Habraken**

   ("Microsoft Office XP 8-in-1" – Part III: Word – Chapter 2: Working with Documents).

   (10) Response to Argument

The Examiner points out that the above rejection has been streamlined with respect to the one presented

on pages 4-10 of the Final Office Action mailed 11/14/2008. Specifically, limitations that were covered

redundantly by both Chikirivao and Wallace are now only recited once, to more clearly focus on how the

secondary reference (Wallace) would modify the primary reference (Chikirivao), and the rationale for

doing so.

Appellant's arguments filed 12/18/2009 have been fully considered but they are not persuasive.

Claim 1:

In re page 5, Appellant argues regarding the claimed "template":

> The Wallace publication simply discloses a mark-up language for inputting knowledge into chat-
> bots. The Wallace publication fails to disclose a "template" and/or a "logic layer" consistent with
> the claims as applied across both the Chikirivao publication and the Wallace publication. The
> Examiner acknowledges on page 5 of the final Office Action how "the term " " in the Wallace
> publication is closer to the claimed "rules" than the claimed "template"." The claims have

limitations encompassing the terms "rules" and "templates", thereby indicating that the terms are literally distinguished from each other, as they serve different roles and embody different concepts. Thus, it can be seen that an inconsistent reading of the disclosure of the Wallace publication is applied to the claims at issue.

Examiner disagrees. Examiner acknowledges that the terminology used by Chikirivao and Wallace differs, as each author functions as a lexicographer for his own work. However, the claimed "template" is taught by both. Appellant does not context the claim mapping to Chikirivao, so Examiner will not treat it further here. Regarding the terminology in Wallace, the examiner has previously established (see the Final Office Action mailed 11/14/2008, especially pages 5, 8, and 18), that Wallace relates to Chikirivao and the claimed limitations such that in Wallace:

- the claimed "template interface" maps to the "template window" on page 42 of Wallace, which is an interface provided to the administrator/botmaster in order to elicit information that will become part of the rules from which a rules-based program provides responses. In other words, the "template window" of Wallace is used as a way to create specific rules based on a predefined form, as required by the instant claims. These rules are then used by a rules-based program to provide responses, as claimed.

- the person of ordinary skill in the art at the time the invention was made would have understood that the information entered into the interface ('template window" on pages 42 of Wallace) in a structured manner for conversion into specific rules, implicitly meets the claimed template.

- the term "<template>" in Wallace reads on the claimed "rules" (rather than the claimed "template"), and these rules are created form the information entered into the "template window" (see Wallace pages 42-44); The Examiner has not equated the "<template>" of Wallace with the claimed template.

Furthermore, the claimed "logic layer" is required to be able to recognize a response and to choose between various responses. Examiner asserts that any branching program logic choosing responses would meet this requirement. For example, in Wallace, a response is recognized as a potential response based on the tags that enclose it -- it is determined to be a valid potential response based on the category and pattern with which it is associated.

In re page 6, Appellant argues:

> Specifically, in §5(a) of his Declaration, Mr. Keane declares:
>> [A]s set forth in the present application, the purpose of a template, is a way to create a specific rule or rules, based on a pre-defined form (the template), containing markers for additional information needed to define the rule (signifiers), as provided by the administrator. In the Wallace publication, AIML tags, which are equated by the Examiner with the Applicants' signifiers, serve a very different role, in that they are only activated during the execution of the AIML rules to control the flow of the program defined by those rules. AIML tags are an exclusively run-time control structure. In contrast, the signifiers in the Applicants' templates, are an exclusively compile-time structure, as they are used to construct run-time rules. There is no disclosure in the Wallace publication with respect to a mechanism that would correspond to a compilation of run-time roles from a partially-defined template. Every feature described in the Wallace publication is part of the run-time system. The use of the term "template", which appears in AIML, means something completely different, as it is the term used to describe the form of an output or reply in a rule. Therefore, it is not the case that AIML tags denote places where rules "need information", as an AIML tag denotes a control branch in the execution of the rule, which may cause recursion, output, or even external code execution.

Examiner asserts that Wallace meets this interpretation, but acknowledges some differences in terminology, as noted above. In Wallace, the administrator/botmaster enters information into a "template window" (Wallace, page 42) which is equivalent to the interface for receiving information from the administrator in the instant invention. The information entered is saved from the template into rules (see Wallace especially "store" page 30 or "save" pages 43-44 or pages 47,50,53 or page 79) to be made accessible to a rules-based program which responds to user requests.

If Appellant is seeking an interpretation of the claimed "template" that is narrower than the reasonable interpretation taken by the Examiner, then Appellant is invited to further define the "template" in the claims in a manner which distinguishes it over the prior art.

As described in the Declaration by Mr. Keane, this involves markers for additional information needed to define the rule (signifiers), specifically the AIML tags which are programming directives.

Regarding the allegation that the AIML tags in Wallace are only activated during execution as a run-time control structure whereas the signifiers in Appellant's templates are exclusively compile-time structures, the Examiner points out that these **arguments are narrower than the claims require.** The claims do <u>not</u> require any features to be "compile-time" features. It is well known in the art to have computer languages that are scripted or interpreted rather than compiled, such as ASP, BASIC, COBOL,

HTML, XML, MATLAB, PHP, Ruby, Smalltalk, etc. Based on this knowledge in the art, the interpretation

taken by the Examiner is extremely reasonable.

If Appellant is seeking coverage for "signifiers" that are compile-time structures, then Appellant is

invited to add language to the claimed limitations to draw that distinction between the claims and the

applied prior art.

In re page 6, Appellant argues:

> Furthermore, on page 18 of the final Office Action, the Examiner asserts that the claimed
> "template" is in fact equated to the disclosed "template window" of the Wallace
> publication. However, even if this interpretation is proper, a consistent reading of this
> equated term would not make sense given that one of the claimed steps would then
> require "retrieving information indicated as needed from a corresponding field in the
> 'template interface'."

The Examiner does not fully understand this argument. As detailed above, the Examiner agrees that

information entered into the "template window" of Wallace is retrieved as needed based on the indication

of a "signifier" (e.g. the AIML tags of Wallace). Examiner fails to see the inconsistency here. What part of

the Examiner's interpretation of the claims is contrary to the functionality of Wallace?

In re pages 6-7, Appellant argues regarding the claimed "signifier":

> The Examiner also equates AIML tags with the "signifiers" set forth in the claims.
> However, equating the two fails to take into account the substantial differences in
> purpose and action between signifiers and AIML tags. Contrary to the Examiner's
> assertion, AIML tags do not indicate that a rule "needs information" as is claimed.
> Instead, AIML tags simply act as organizing directives, as supported by the disclosure
> of the Wallace publication on page 12 reprinted below:
>> More generally, AIML tags transform the reply into a mini computer program
>> which can save data, activate other programs, give conditional responses, and
>> recursively call the pattern matcher to insert the responses from other categories.
>> Most AIML tags in fact belong to this template side sub language.
> The Wallace publication refers to tags as a "sub language". AIML tags do not have the
> property that causes the information to be "linked to the rule", as is required by the
> claims. The signifiers of the present invention are markers that identify a specific piece of
> information "such that the call for information invokes a process to select the information
> from a corresponding field in the template so that the information will be linked to the rule"
> as is set forth in the claims.
>> On page 19, lines 9-14 of the final Office Action, the Examiner asserts how "[t]he
>> AIML tags may indicate information is needed from another function (<srai>) or by picking
>> from a finite set of choices (<random>). That <random> indicates choosing randomly
>> does not indicate any lack of logic, so long as the random selection is between equally
>> valid responses." Specifically, the Examiner attempts to rationalize how a random
>> selection due to the "<random>" AIML tag can be interpreted as being the most logical
>> choice, by stating that "a random selection is the most logical choice, given that there is

> no logical difference between the responses". Appellants fail to understand how logical
> decision making, as is claimed, is anticipated by this rationale.

Examiner disagrees. It is true that the examiner has equated AIML tags with the claimed "signifier".

Examiner maintains that these AIML tags do indicate that information is needed from elsewhere. AIML

tags explicitly indicate that a rule (e.g. a "<template>" in Wallace) needs information. The <srai> function,

which indicates that more information (not currently in the rule) is needed from another rule or function is

explicitly a signification that more information is needed.

Though they are programming directives or a "sub language", they meet the requirements of the

claim limitations. Linking information to the rule, in the broadest reasonable sense, is establishing **any**

connection between the rule and the information. The AIML tags do this by indicating it is appropriate to

have a conditional response, indicating a recursive call needs to be made, etc. Programming directives

such as those detailed in Wallace are reasonably interpreted to signify that a rule needs more

information, as claimed.

In the quotation above from page 7, Appellant states that the "signifiers of the present invention

are markers that identify as specific piece of information 'such that the call for information invokes a

process to select the information from a corresponding field in the template so that the information will be

linked to the rule". The Examiner points out that, for example, AIML tags can be used to pull in specific

information to reduce complex grammar to something simpler to understand, can be use to retrieve

synonyms for particular terms to help rephrase a response, can be used to detect and respond to

keywords, and much more (e.g. see Wallace page 13). This specific information, initially entered by the

administrator/botmaster through the template window, is linked to the rule being processed to create

meaningful conversation.

Regarding the <random> tag, the Examiner does not see how Appellant's argument in any way

contradicts the interpretation of <random> as a signifier that calls for information entered into a particular

structure of the template window. Examiner also fails to see how Appellant's argument in any way

disputes that response chosen randomly from a particular set would be reasonably understood to be the

most logical in situations when one particular choice is not measurably superior to another.

Understanding that Wallace is seeking to have believable conversation with a user, the Examiner asserts

that the person of ordinary skill in the art at the time the invention was made would have clearly

understood that this type of unpredictability would be logically required.

      Also, the Examiner points out that this is not an anticipation rejection, but rather a rejection under

35 U.S.C. §103 showing that the claims are obvious with respect to the cited prior art (Chikirivao and

Wallace).

      In re pages 7-8, Appellant argues regarding the claimed "logic layer":

>   With respect to the "logic layer" limitation of the claims, Mr. Keane declares in §5(b) of his
> Declaration:
>> The logic layer of the present invention implements a distinct processing step whereby a
>> specific response can be selected from a set of equivalent responses in a principled
>> manner, which may reference information not present in the user input to the system.
>> Neither the Chikirivao nor Wallace publications show this feature. The Wallace
>> publication describes AIML, however, AIML is a language without the explicit notion of
>> the claimed logic layer, as responses in AIML proceed directly from rule input matching,
>> or recursion on the input matching. It is incorrect to equate the "Graphmaster" of the
>> Wallace publication to the claimed logic layer because the Graphmaster is a
>> representation of the input space matching capabilities of a particular AIML definition set
>> (e.g., A.L.I.C.E.). There is no explicit logic layer in AIML, as the output is determined by
>> the user input and the rules. In the present invention, the matching of an input is only the
>> first step to determining the appropriate response, followed by the processing of the logic
>> layer.
> The AIML model does not have an explicit logic layer, as found in the present invention. The tag
> grouping in AIML allows an implicit variation among equivalent outputs through random choice.
> There is no disclosure, teaching, or suggestion in the Wallace publication of a mechanism
> equivalent to the claimed logic layer that is able to take additional information, if needed, and
> perform a defined computation that can determine the selection of an output from among a set of
> outputs.

Examiner disagrees. The broadest reasonable interpretation of the claim language certainly doesn't

require that the logic layer *understand* the possible responses. Wallace discusses how the AIML

interpreter is capable of logically deducing appropriate responses based on patterns and associations

(Wallace, i.e. page 55) – thereby filling in needed information through inference as guided by the tags.

Appellant's arguments neglect to consider that every quest for an answer begins with every piece of

knowledge in the system, which AIML refines based on a tracked category, then by pattern matching.

These are all logical operations which select an answer among many possibilities. That a final selection

may be made randomly is no less logical so long as the possible answers from which the selection is

made are equally valid. That output is determined by subjecting user inputs to a set of rules is evidence

of a logical processing of the input by a layer of AIML -- thus the claimed "logic layer" is met.

The claimed "logic layer" is required to be able to recognize a response and to choose between

various responses. Examiner asserts that any branching program logic choosing responses would meet

this requirement. For example, in Wallace, a response is recognized as a potential response based on

the tags that enclose it -- it is determined to be a valid potential response based on the category and

pattern with which it is associated. If Appellant desires a logic layer that is not reasonably interpreted as

a pattern matcher, then Appellant is invited to add claim limitations to distinguish over the cited prior art in

this manner.

In re page 8, Appellant argues:

> In AIML, the tag allows for a recursive rule definition (See page 13 of the Wallace
> publication). While this allows considerable flexibility in the form of rule definition, it does
> not, in and of itself, introduce any functionality in the matching power of AIML that could
> not be replicated by finite-length, non-recursive matching rules. The tag does not provide
> the concept of choice over several possible outcomes all appropriate for a particular
> input, but selectable by additional information not contained in the input.

Examiner disagrees. AIML can accomplish non-recursive matching through, for example, conditional

branching, targeting, and context (e.g. Wallace pages 16-20). Examiner doesn't see any claimed

limitations related to whether or not the rules are recursive or not recursive. The AIML tags provide

choice over a plurality of appropriate responses based on context, targets, wildcards, deductions,

reductions, inferences, and signifiers (AIML tags) that logically branch and retrieve information (as

described above) and randomly choose when appropriate (e.g. see Wallace pages 35-36 or pages 41-44

or pages 54-57).

In re page 8, Appellant argues:

> With respect to the "logic layer", the Examiner now broadly construes this term in the final
> Office Action to any branching program logic. However, the claim requires that the logic
> layer be configured to choose between various responses provided by the administrator.
> Thus, just because a response is recognized (i.e., identified) as a "response" in the
> Wallace publication, this action cannot be equated to the active step of making a choice
> between responses.

Examiner disagrees. For clarity, the Examiner concedes that the branching program logic needs to

choose responses provided by the administrator. However, the Examiner asserts that the responses

selected as appropriate by Wallace are entered by the administrator, as established above, and that AIML

does choose logically among such administrator entered information based on context, targets, wildcards,

deductions, reductions, inferences, and signifiers (e.g. see Wallace pages 35-36 or pages 41-44 or pages

54-57), as established above.

In re pages 8, Appellant argues:

> Accordingly, the claimed concept of a "template" and "logic layer" in the context of a
> consistent reading of the other limitations of the independent claims and the claims
> depending therefrom is not disclosed, taught, or suggested in any of the prior art of
> record. In light of the aforementioned arguments made with respect to the anticipation
> rejections under the Wallace publication, whose underlying anticipation teachings, now
> refuted, are used for rejecting at least the independent claims on an obviousness basis in
> view of the teaching of the Chikirivao publication, Appellants hereby respectfully request
> that the Examiner's obviousness rejection be reversed.

Examiner disagrees. The Examiner again points out that this is **not** an anticipation rejection, but rather a

rejection under 35 U.S.C. §103 showing that the claims are obvious with respect to Chikirivao and

Wallace. The standard is not anticipation of the claimed limitations, but rather obviousness of the claimed

limitations over the cited art. As detailed above, the claimed "template", "signifier", and "logic layer" are

taught by Wallace in a context consistent with the claims, and when combined to modify Chikirivao, all

claimed limitations are taught as detailed in the rejection above. Therefore, this outstanding rejection

under 35 U.S.C. §103 is properly MAINTAINED.

In re pages 8-9, Appellant argues:

> When undertaking an obviousness analysis, the Examiner is also required to take into
> account secondary considerations relation to applicant's invention. Graham v. John
> Deere, 383 U.S. 1, 148 USPQ 459 (1966). The Court of Appeals for the Federal Circuit
> stated in Stratoflex, Inc. v. Aeroquip Corp., 713 F.2d 1530, 1538, 218 USPQ 871,879
> (Fed. Cir. 1983) that "evidence rising out of the so-called 'secondary considerations',
> such as commercial success, must always when present be considered en route to a
> determination of obviousness" (See also KSR v. Teleflex, 550 U.S. ___ at 2). In §3 of his
> Declaration, Mr. Keane:
>> The claimed invention was first commercialized in July of 2003. The Assignee
>> has commercially pursued deployments that incorporate the claimed invention.
>> With respect to enterprise deployments, the Assignee [has] offered, since as
>> early as 2005 and through the present, implementations containing the claimed
>> invention to various companies including PSEG (a major Northeastern U.S.
>> electric and gas utility company), Qantas Airlines (the leading Australian air
>> carrier), and the University of Phoenix (one of the largest higher educational
>> institutions in the world, enrolling approximately 400,000 students). Currently, the
>> Assignee is in contract negotiations to deploy its product through Sutherland
>> Global Services, a company providing business process outsourcing services to

Fortune 500 companies, including Dell. The estimated total sales of products and deployments that incorporate the claimed invention are at least $150,000. The foregoing information is indicative of the commercial success of the claimed invention.

As set forth in §4 of the Declaration, customers utilizing the claimed invention reduce their costs while improving their online customer service. As discussed in the Declaration, as an example, cumbersome manual processes have been replaced with automation-assisted online processes, thereby reducing errors and increasing customer satisfaction.

In the final Office Action, the Examiner asserts that the submission of the Declaration, has "not proven [that] the commercial success is [a] result of a direct correspondence with the claimed 'template' and/or 'logic layer' and/or 'signifier'." Contrary to the Examiner's understanding, a §1.132 Declaration is not required to address any particular claim limitations, but may be focused on the claimed invention as a whole (e.g., the benefit obtained from practicing what is claimed). In other words, the statements relating to secondary considerations in a § 1.132 Declaration are not intended to go toward a showing of a correspondence between commercial success and how any one specific claim limitation relates to achieving that commercial success. Rather, the § 1.132 Declaration is used to show that, assuming, arguendo, that an Examiner's obviousness combination may be defensible in light of one's arguments against obviousness, the fact that commercial success has been achieved from commercially implementing what is being claimed, is indicative of the value of the invention, thereby, overriding the §103 rejection. By dismissing a portion of the Declaration on improper procedural grounds, the Examiner has failed to address the merits of commercial success of the claimed invention, as discussed by Appellants, above.

Examiner disagrees. The Examiner **has considered** the evidence rising out of the so-called "secondary considerations", but has **not** found it to be persuasive. Applicant has **not** proven that the commercial success is result of a direct correspondence with the claimed "template" and/or "logic layer" and/or "signifier" and/or the **combination** of limitation found in the independent claims. The Declaration is directed to the invention as a whole, which includes many features that are found in the disclosure, but not in the claims. As Appellant has pointed out on pages 2-3 and 10 of the instant Appeal Brief, the present invention, as disclosed in the specification, is directed at 'enabling administrators with no specialized background, training, or expertise to enter information that will be incorporated into patterns an used by a Bot, thereby enabling the Bot to correctly respond to a multitude of different questions with the information the administrator provides." Furthermore, the specification similarly state that the "this invention is directed to a method and system for the mass customization of Bots by administrators having no specialized knowledge or experience in the art" (¶[0001] on page 1 of the specification as originally filed) and "there is a need in the art to have a method of scripting a Bot which successfully customizes the Bot to an individual application using knowledge that can be given to the Bot by any lay person or low-

level employee" (¶[0007] on page 3 of the specification as originally filed) and "the present invention …

[provides] a method of creating Bots according to patterns (or text strings) that are written in a very high

level language that closely resembles a human natural language" (¶[0008] on page 3 of the specification

as originally filed). The Examiner does not see any claim limitations directed to the experience level,

training, or training of the administrator. The Examiner further does not see any claim limitations related

to "mass customization", "human natural language", or customization to "an individual application". The

Examiner reasonably interprets unclaimed features of the invention as a whole as being at least partially

responsible for the commercial success of the invention as a whole, in light of the full disclosure and

consideration of the Declaration.

As the commercial success is not explicitly tied to the **specific combination of limitations**

claimed the examiner is **not** persuaded that any of these features are directly responsible for the

commercial success. Therefore, the commercial success **has** been considered, but is **not** persuasive.

Therefore, this outstanding rejection under 35 U.S.C. §103 is properly MAINTAINED.

Claim 35:

In re page 10, Appellant argues:

> The Final Office Action dated November 14, 2008 (hereinafter "the final Office Action")
> fails to set forth a proper obviousness rejection with respect to claim 35, as the Chikirivao
> and Wallace references taken in combination fail to anticipate every limitation of claims
> 35. Appellants offer the same arguments herein with respect to the rejection of claim 35,
> as have been set forth above with respect to claim 1, above. Appellants hereby
> incorporate by reference the arguments made above with respect to the requirement of
> having the Examiner properly consider the secondary considerations set forth in the
> Declaration.

Examiner disagrees. The Examiner again points out that this is **not** an anticipation rejection, but rather a

rejection under 35 U.S.C. §103 showing that the claims are obvious with respect to Chikirivao and

Wallace. The standard is not anticipation of the claimed limitations, but rather obviousness of the claimed

limitations over the cited art. Appellant argues that claim 35 is patentable for the same reasons as

presented for claim 1 above. The Examiner has shown in detail above that Appellant's arguments are not

persuasive. This outstanding rejection of claim 35 under 35 U.S.C. §102 is properly MAINTAINED for the

same reasons given for claim 1 above.

Conclusion

In re page 10, Appellant argues:

> The present invention is a novel and non-obvious way of enabling administrators with no specialized background, training, or expertise to enter information that will be incorporated into patterns and used by a Bot, thereby enabling the Bot to correctly respond to a multitude of different questions with the information the administrator provides.
> It is settled law that "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Brothers Inc. v. Union Oil Co. of California, 2 USPQ2d, 1051, 1053 (Fed. Cir. 1987). In light of the aforementioned arguments made with respect to the anticipation rejections under the Wallace publication, whose underlying anticipation teachings, are used for rejecting at least claims 1 and 35 on an obviousness basis in view of the teaching of the Chikirivao publication, Appellants hereby respectfully request that the Examiner's improper obviousness rejection be reversed.

Examiner disagrees. Appellant's arguments are narrower than the claims require: there are no limitations drawn to the background, training, or expertise of the administrator. The examiner once again points out that this is **not** an anticipation rejection, but rather a rejection under 35 U.S.C. §103 showing that the claims are obvious with respect to Chikirivao and Wallace. The standard is not anticipation of the claimed limitations, but rather obviousness of the claimed limitations over the cited art. All of Appellant's arguments have been fully considered, but they are not persuasive. The examiner has shown that the claims are properly rejected under 35 U.S.C. 103(a) as being unpatentable over the combination of **Chikirivao** (USPAP 2003/0163783) and **Wallace** ("The Elements of AIML Style"), as detailed above. All outstanding claim rejections are properly MAINTAINED.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Benjamin Buss 2/13/2010

/B. B./

Examiner, Art Unit 2129

Conferees:

/Donald Sparks/

Supervisory Patent Examiner, Art Unit 2129

/Eddie C. Lee/

Quality Assurance Specialist